Lecture 3: Quantifiers and Counting

Jakub Szymanik

university of groningen

Outline

Outline

Languages – Basic Definitions

- *Alphabet* is any non-empty finite set of symbols, e.g., $A = \{a, b\}$ and $B = \{0, 1\}$.

Languages – Basic Definitions

- *Alphabet* is any non-empty finite set of symbols, e.g., $A = \{a, b\}$ and $B = \{0, 1\}$.
- A *word (string)* is a finite sequence of symbols from a given alphabet, e.g., “1110001110”.

Languages – Basic Definitions

- *Alphabet* is any non-empty finite set of symbols, e.g., $A = \{a, b\}$ and $B = \{0, 1\}$.
- A *word (string)* is a finite sequence of symbols from a given alphabet, e.g., "1110001110".
- The *empty word*, ε, is a sequence without symbols.

Languages – Basic Definitions

- *Alphabet* is any non-empty finite set of symbols, e.g., $A = \{a, b\}$ and $B = \{0, 1\}$.
- A *word (string)* is a finite sequence of symbols from a given alphabet, e.g., “1110001110”.
- The *empty word*, ε, is a sequence without symbols.
- The *length of a word* is the number of symbols in it.

Languages – Basic Definitions

- *Alphabet* is any non-empty finite set of symbols, e.g., $A = \{a, b\}$ and $B = \{0, 1\}$.
- A *word (string)* is a finite sequence of symbols from a given alphabet, e.g., "1110001110".
- The *empty word*, ε, is a sequence without symbols.
- The *length of a word* is the number of symbols in it.
- *The set of all words over alphabet* Γ is denoted by Γ^*, e.g., $\{0, 1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$.

Languages – Basic Definitions

- *Alphabet* is any non-empty finite set of symbols, e.g., $A = \{a, b\}$ and $B = \{0, 1\}$.
- A *word (string)* is a finite sequence of symbols from a given alphabet, e.g., “1110001110”.
- The *empty word*, ε, is a sequence without symbols.
- The *length of a word* is the number of symbols in it.
- *The set of all words over alphabet* Γ is denoted by Γ^*, e.g., $\{0, 1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$.
- Any set of words, a subset of Γ^*, will be called a *language*.

Finite Automata

Definition

A *non-deterministic finite automaton* (FA) is a tuple (A, Q, q_s, F, δ), where:

- A is an input alphabet;
- Q is a finite set of states;
- $q_s \in Q$ is an initial state;
- $F \subseteq Q$ is a set of accepting states;
- $\delta : Q \times A \longrightarrow \mathcal{P}(Q)$ is a transition function.

Regular Languages

Definition

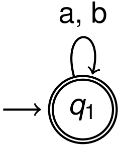
The language accepted (recognized) by some FA H, L(H), is the set of all words over the alphabet A which are accepted by H.

Definition

We say that a *language $L \subseteq A^*$ is regular* if and only if there exists some FA H such that $L = L(H)$.

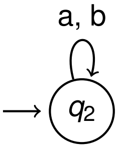
Example 1

Let $A = \{a, b\}$ and consider the language $L_1 = A^*$.



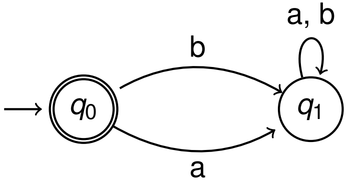
Example 2

Let $L_2 = \emptyset$



Example 3

$L_3 = \{\varepsilon\}$



Not Every Language is Regular

$$L_{ab} = \{a^n b^n : n \geq 1\}$$

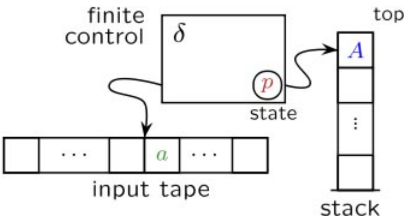
Push Down Automata

Definition

A non-deterministic push-down automaton (PDA) is a tuple $(A, \Gamma, \#, Q, q_s, F, \delta)$, where:

- A is an input alphabet;
- Γ is a stack alphabet;
- $\# \notin \Gamma$ is a stack initial symbol, empty stack consists only from it;
- Q is a finite set of states;
- $q_s \in Q$ is an initial state;
- $F \subseteq Q$ is a set of accepting states;
- $\delta : Q \times (A \cup \{\varepsilon\}) \times \Gamma \longrightarrow \mathcal{P}(Q \times \Gamma^*)$ is a transition function.

PDA



push/pop-off a symbol from the top of the stack

Context-free Languages

Definition

We say that a language $L \subseteq A^*$ is context-free if and only if there is a PDA H such that $L = L(H)$.

Regular $\subset$ Context-free

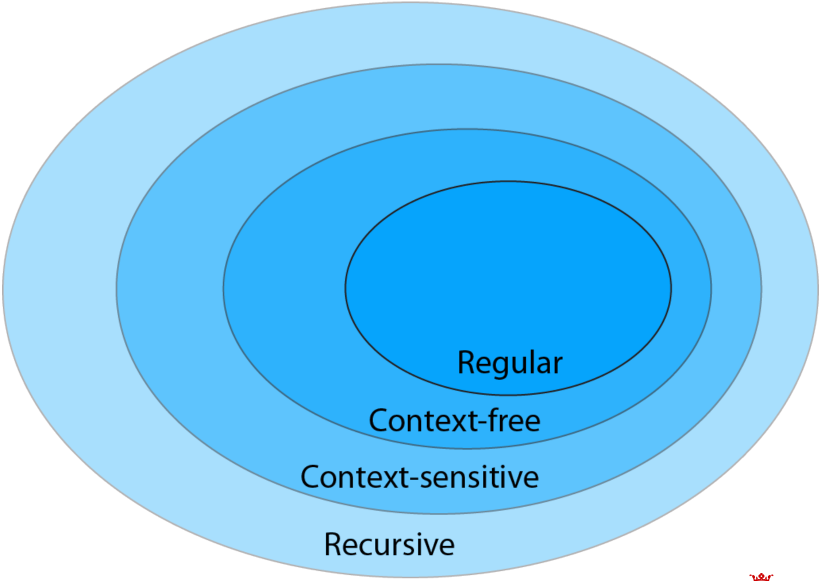
There is a PDA for $L_{ab} = \{a^n b^n : n \geq 1\}$.

Beyond Context-free Languages

$$L_{abc} = \{a^k b^k c^k : k \geq 1\}$$

We will investigate stronger languages in the last lecture.

Chomsky's Hierarchy



Outline

Lindström Definition

Definition

Let $t = (n_1, \ldots, n_k)$ be a k-tuple of positive integers. *A generalized quantifier* of type t is a class Q of models of a vocabulary $\tau_t = \{R_1, \ldots, R_k\}$, such that R_i is n_i-ary for $1 \leq i \leq k$, and Q is closed under isomorphisms, i.e. if $\mathbb{M}$ and $\mathbb{M}'$ are isomorphic, then

$$(\mathbb{M} \in \mathsf{Q} \iff \mathbb{M}' \in \mathsf{Q}).$$

Lindström Definition

Definition

Let $t = (n_1, \ldots, n_k)$ be a k-tuple of positive integers. *A generalized quantifier* of type t is a class Q of models of a vocabulary $\tau_t = \{R_1, \ldots, R_k\}$, such that R_i is n_i-ary for $1 \leq i \leq k$, and Q is closed under isomorphisms, i.e. if $\mathbb{M}$ and $\mathbb{M}'$ are isomorphic, then

$$(\mathbb{M} \in \mathsf{Q} \iff \mathbb{M}' \in \mathsf{Q}).$$

Definition

If in the above definition for all i: $n_i = 1$, then we say that a quantifier is *monadic*, otherwise we call it *polyadic*.

Simple Quantifier Sentences

- Every poet has low self-esteem.
- Some dean danced nude on the table.
- At least 3 grad students prepared presentations.
- An even number of the students saw a ghost.
- Most of the students think they are smart.
- Less than half of the students received good marks.

Monadic Quantifiers of Type (1, 1)

Definition

A monadic generalized quantifier of type (1,1) is a class Q of structures of the form $M = (U, A_1, A_2)$, where $A_1, A_2 \subseteq U$. Additionally, Q is closed under isomorphism.

Examples

$$\text{every} = \{(M, A, B) \mid A, B \subseteq M \text{ and } A \subseteq B\}.$$

Examples

$$\text{every} = \{(M, A, B) \mid A, B \subseteq M \text{ and } A \subseteq B\}.$$

$$\text{some} = \{(M, A, B) \mid A, B \subseteq M \text{ and } A \cap B \neq \emptyset\}.$$

Examples

$$\text{every} = \{(M, A, B) \mid A, B \subseteq M \text{ and } A \subseteq B\}.$$

$$\text{some} = \{(M, A, B) \mid A, B \subseteq M \text{ and } A \cap B \neq \emptyset\}.$$

$$\text{more than k} = \{(M, A, B) \mid A, B \subseteq M \text{ and } \text{card}(A \cap B) > k\}.$$

Examples

$$\text{every} = \{(M, A, B) \mid A, B \subseteq M \text{ and } A \subseteq B\}.$$

$$\text{some} = \{(M, A, B) \mid A, B \subseteq M \text{ and } A \cap B \neq \emptyset\}.$$

$$\text{more than k} = \{(M, A, B) \mid A, B \subseteq M \text{ and } \mathrm{card}(A \cap B) > k\}.$$

$$\text{even} = \{(M, A, B) \mid A, B \subseteq M \text{ and } \mathrm{card}(A \cap B) \text{ is even}\}.$$

Examples

$$\text{every} = \{(M, A, B) \mid A, B \subseteq M \text{ and } A \subseteq B\}.$$

$$\text{some} = \{(M, A, B) \mid A, B \subseteq M \text{ and } A \cap B \neq \emptyset\}.$$

$$\text{more than k} = \{(M, A, B) \mid A, B \subseteq M \text{ and } \mathrm{card}(A \cap B) > k\}.$$

$$\text{even} = \{(M, A, B) \mid A, B \subseteq M \text{ and } \mathrm{card}(A \cap B) \text{ is even}\}.$$

$$\text{most} = \{(M, A, B) \mid A, B \subseteq M \text{ and } \mathrm{card}(A \cap B) > \mathrm{card}(A - B)\}$$

2 Concepts of GQ are Equivalent

$$(M, A_1, A_2) \in \mathrm{Q} \iff \mathrm{Q}_M(A_1, A_2), \text{where } A_i \subseteq M, \; i = 1, 2.$$

Example

$$(M, A_1, A_2) \in \mathrm{most} \iff \mathrm{most}_M(A_1, A_2).$$

2 Concepts of GQ are Equivalent

$$(M, A_1, A_2) \in \mathrm{Q} \iff \mathrm{Q}_M(A_1, A_2), \text{where } A_i \subseteq M,\ \ i = 1, 2.$$

Example

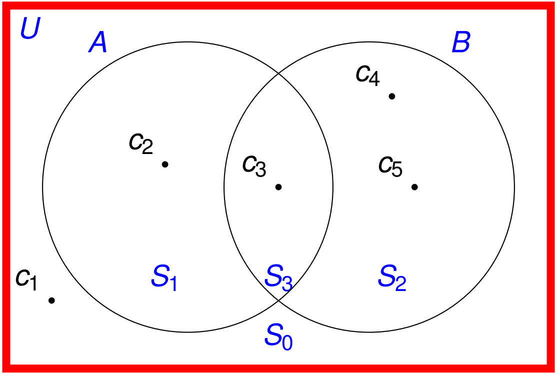
$$(M, A_1, A_2) \in \text{most} \iff \text{most}_M(A_1, A_2).$$

Corollary

The two definitions of generalized quantifiers are equivalent.

Outline

How do We Encode Models?



This model is uniquely described by $\alpha_M = a_{\bar{A}\bar{B}} a_{A\bar{B}} a_{AB} a_{\bar{A}B} a_{\bar{A}B}$

Step by Step

- Restriction to finite models of the form $M = (U, A, B)$.

Step by Step

- Restriction to finite models of the form $M = (U, A, B)$.
- List of all elements of the model: $c_1, \ldots, c_5$.

Step by Step

- Restriction to finite models of the form $M = (U, A, B)$.
- List of all elements of the model: $c_1, \ldots, c_5$.
- Labeling every element with one of the letters: $a_{\bar{A}\bar{B}}$, $a_{A\bar{B}}$, $a_{\bar{A}B}$, a_{AB}, according to constituents it belongs to.

Step by Step

- Restriction to finite models of the form $M = (U, A, B)$.
- List of all elements of the model: $c_1, \ldots, c_5$.
- Labeling every element with one of the letters: $a_{\bar{A}\bar{B}}$, $a_{A\bar{B}}$, $a_{\bar{A}B}$, a_{AB}, according to constituents it belongs to.
- Result: the word $\alpha_M = a_{\bar{A}\bar{B}} a_{A\bar{B}} a_{AB} a_{\bar{A}B} a_{\bar{A}B}$.

Step by Step

- Restriction to finite models of the form $M = (U, A, B)$.
- List of all elements of the model: $c_1, \ldots, c_5$.
- Labeling every element with one of the letters:
  $a_{\bar{A}\bar{B}}$, $a_{A\bar{B}}$, $a_{\bar{A}B}$, a_{AB}, according to constituents it belongs to.
- Result: the word $\alpha_M = a_{\bar{A}\bar{B}} a_{A\bar{B}} a_{AB} a_{\bar{A}B} a_{\bar{A}B}$.
- α_M describes the model in which:
  $c_1 \in \bar{A}\bar{B}, c_2 \in A\bar{B} c_3 \in AB, c_4 \in \bar{A}B, c_5 \in \bar{A}B$.

Step by Step

- Restriction to finite models of the form $M = (U, A, B)$.
- List of all elements of the model: $c_1, \ldots, c_5$.
- Labeling every element with one of the letters: $a_{\bar{A}\bar{B}}$, $a_{A\bar{B}}$, $a_{\bar{A}B}$, a_{AB}, according to constituents it belongs to.
- Result: the word $\alpha_M = a_{\bar{A}\bar{B}} a_{A\bar{B}} a_{AB} a_{\bar{A}B} a_{\bar{A}B}$.
- α_M describes the model in which: $c_1 \in \bar{A}\bar{B}, c_2 \in A\bar{B} c_3 \in AB, c_4 \in \bar{A}B, c_5 \in \bar{A}B$.
- The class Q is represented by the set of words describing all elements of the class.

Constituents – General Definition

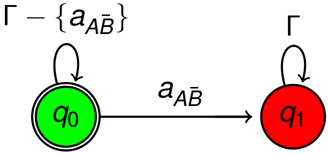
Definition

The class K_{Q} of finite models of the form $(M, A_1, \ldots, A_n)$ can be represented by the set of nonempty words L_{Q} over the alphabet $A = \{a_1, \ldots, a_{2^n}\}$ such that: $\alpha \in L_{\mathrm{Q}}$ if and only if there are $(U, A_1, \ldots, A_n) \in K_{\mathrm{Q}}$ and linear ordering $U = \{c_1, \ldots, c_k\}$, such that $length(\alpha) = k$ and i-th character of α is a_j exactly when $c_i \in S_1 \cap \ldots \cap S_n$, where:

$$S_l = \begin{cases} A_l & \text{if integer part of } \frac{j}{2^l} \text{ is odd} \\ U - A_l & \text{otherwise.} \end{cases}$$

Aristotelian Quantifiers

"all", "some", "no", and "not all"

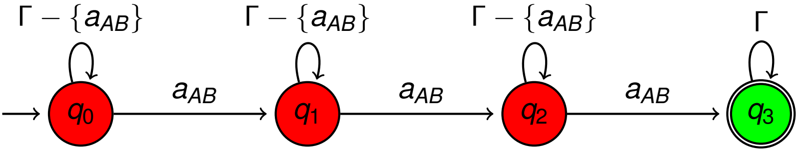


Finite automaton recognizing L_{All}

$$L_{\text{All}} = \{\alpha \in \Gamma^* : \# a_{A\bar{B}}(\alpha) = 0\}$$

Cardinal Quantifiers

E.g. "more than 2", "less than 7", and "between 8 and 11"

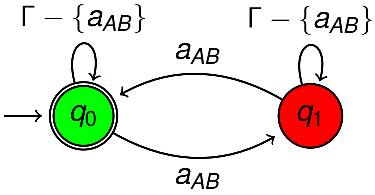


Finite automaton recognizing $L_{\text{More than two}}$

$$L_{\text{More than two}} = \{\alpha \in \Gamma^* : \#a_{AB}(\alpha) > 2\}$$

Parity Quantifiers

E.g. "an even number", "an odd number"



Finite automaton recognizing L_{Even}

$$L_{\text{Even}} = \{\alpha \in \Gamma^* : \#a_{AB}(\alpha) \text{ is even}\}$$

Proportional Quantifiers

- E.g. “most”, “less than half”.
- Most *As are B* iff card$(A \cap B) > $ card$(A - B)$.
- $L_{\text{Most}} = \{\alpha \in \Gamma^* : \#a_{AB}(\alpha) > \#a_{A\bar{B}}(\alpha)\}$.
- There is no finite automaton recognizing this language.
- We need internal memory.
- A push-down automata will do.

university of groningen

Correspondence

Question

What does it mean that a class of monadic quantifiers is recognized by a class of devices?

Correspondence

Question

What does it mean that a class of monadic quantifiers is recognized by a class of devices?

Definition

Let $\mathcal{D}$ be a class of recognizing devices,
Ω a class of monadic quantifiers.
We say that $\mathcal{D}$ accepts Ω if and only if
for every monadic quantifier Q:

$$\mathsf{Q} \in \Omega \iff \text{there is device } A \in \mathcal{D}(A \textit{ accepts } L_{\mathsf{Q}}).$$

Relevant Results: Acyclic FA and FA

Theorem (J. van Benthem)

Monadic quantifier Q *is first–order definable iff* L_Q *is accepted by an acyclic finite automaton.*

Relevant Results: Acyclic FA and FA

Theorem (J. van Benthem)

Monadic quantifier Q *is first–order definable iff* L_{Q} *is accepted by an acyclic finite automaton.*

Theorem (M. Mostowski)

Monadic quantifier Q *is definable in the divisibility logic iff* L_{Q} *is accepted by a finite automaton.*

Relevant Results: Acyclic FA and FA

Theorem (J. van Benthem)

Monadic quantifier Q *is first–order definable iff* L_{Q} *is accepted by an* acyclic *finite automaton.*

Theorem (M. Mostowski)

Monadic quantifier Q *is definable in the divisibility logic iff* L_{Q} *is accepted by a finite automaton.*

FA do not use any kind of working memory device.

Odds of “Even”

- “Even” and “odd” are non-FO.
- They can be however recognized by FA.
- But opposite to FO quantifiers you need FA with cycle.
- Difference between FA and acyclic FA.

Relevant Qesults: Quantifiers and PDAs

Theorem (J. van Benthem)

Quantifier Q *of type (1) is semilinear iff*
L_Q *is accepted by push–down automaton.*

Relevant Qesults: Quantifiers and PDAs

Theorem (J. van Benthem)

Quantifier Q *of type (1) is semilinear iff*
L_{Q} *is accepted by push–down automaton.*

PDA use stack which is simple working memory device.

Summing Up

Definability	Examples	Recognized by
FO	“all” “at least 3”	acyclic FA
FO(D_n)	“an even number”	FA
PrA	“most”, “less than half”	PDA

Quantifiers, definability, and complexity of automata

Van Benthem, Essays in logical semantics, 1986.

Mostowski, Computational semantics for monadic quantifiers, 1998.

Does it Say Anything About Processing?

Question

Do minimal automata predict differences in verification?

Does it Say Anything About Processing?

Question

Do minimal automata predict differences in verification?

We'll try to convince you that the answer is positive!

Outline

Predictions

- RT will increase along with the computational resources.

Predictions

- RT will increase along with the computational resources.
- Aristotelian qua. $<$ parity qua. $<$ proportional qua.

Predictions

- RT will increase along with the computational resources.
- Aristotelian qua. < parity qua. < proportional qua.
- Aristotelian qua. < cardinal qua. of high rank.

Participants

- 40 native Polish-speaking adults (21 female).
- Volunteers: undergraduates from the University of Warsaw.
- The mean age: 21.42 years (SD = 3.22).
- Each participant tested individually.

Materials

80 grammatically simple propositions in Polish, like:

1. Some cars are red.
2. More than 7 cars are blue.
3. An even number of cars is yellow.
4. Less than half of the cars are black.

Materials Continued

More than half of the cars are yellow.

An example of a stimulus used in the first study

Procedure

- 8 different quantifiers divided into four groups.

Procedure

- 8 different quantifiers divided into four groups.
  - “all” and “some” (acyclic 2-state FA);

Procedure

- 8 different quantifiers divided into four groups.
  - "all" and "some" (acyclic 2-state FA);
  - "odd" and "even" (2-state FA);

Procedure

- 8 different quantifiers divided into four groups.
  - “all” and “some” (acyclic 2-state FA);
  - “odd” and “even” (2-state FA);
  - “less than 8” and “more than 7” (FA);

Procedure

- 8 different quantifiers divided into four groups.
  - “all” and “some” (acyclic 2-state FA);
  - “odd” and “even” (2-state FA);
  - “less than 8” and “more than 7” (FA);
  - “less than half” and “more than half” (PDA).

Procedure

- 8 different quantifiers divided into four groups.
  - “all” and “some” (acyclic 2-state FA);
  - “odd” and “even” (2-state FA);
  - “less than 8” and “more than 7” (FA);
  - “less than half” and “more than half” (PDA).
- Each quantifier was presented in 10 trials.

Procedure

- 8 different quantifiers divided into four groups.
  - “all” and “some” (acyclic 2-state FA);
  - “odd” and “even” (2-state FA);
  - “less than 8” and “more than 7” (FA);
  - “less than half” and “more than half” (PDA).
- Each quantifier was presented in 10 trials.
- The sentence true in the picture in half of the trials.

Procedure

- 8 different quantifiers divided into four groups.
  - “all” and “some” (acyclic 2-state FA);
  - “odd” and “even” (2-state FA);
  - “less than 8” and “more than 7” (FA);
  - “less than half” and “more than half” (PDA).
- Each quantifier was presented in 10 trials.
- The sentence true in the picture in half of the trials.
- Quantity of target items near the criterion of validation.

Procedure

- 8 different quantifiers divided into four groups.
  - “all” and “some” (acyclic 2-state FA);
  - “odd” and “even” (2-state FA);
  - “less than 8” and “more than 7” (FA);
  - “less than half” and “more than half” (PDA).
- Each quantifier was presented in 10 trials.
- The sentence true in the picture in half of the trials.
- Quantity of target items near the criterion of validation.
- Proportional and cardinal statement equivalent

Procedure

- 8 different quantifiers divided into four groups.
  - “all” and “some” (acyclic 2-state FA);
  - “odd” and “even” (2-state FA);
  - “less than 8” and “more than 7” (FA);
  - “less than half” and “more than half” (PDA).
- Each quantifier was presented in 10 trials.
- The sentence true in the picture in half of the trials.
- Quantity of target items near the criterion of validation.
- Proportional and cardinal statement equivalent
- Practice session followed by the experimental session.

Procedure

- 8 different quantifiers divided into four groups.
  - “all” and “some” (acyclic 2-state FA);
  - “odd” and “even” (2-state FA);
  - “less than 8” and “more than 7” (FA);
  - “less than half” and “more than half” (PDA).
- Each quantifier was presented in 10 trials.
- The sentence true in the picture in half of the trials.
- Quantity of target items near the criterion of validation.
- Proportional and cardinal statement equivalent
- Practice session followed by the experimental session.
- Each quantifier problem was given one 15.5 s event.

Procedure

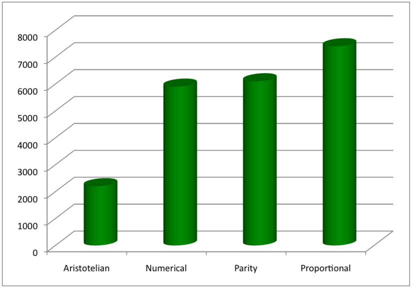
- 8 different quantifiers divided into four groups.
  - “all” and “some” (acyclic 2-state FA);
  - “odd” and “even” (2-state FA);
  - “less than 8” and “more than 7” (FA);
  - “less than half” and “more than half” (PDA).
- Each quantifier was presented in 10 trials.
- The sentence true in the picture in half of the trials.
- Quantity of target items near the criterion of validation.
- Proportional and cardinal statement equivalent
- Practice session followed by the experimental session.
- Each quantifier problem was given one 15.5 s event.
- Subjects were asked to decide the truth-value.

Analysis of Accuracy

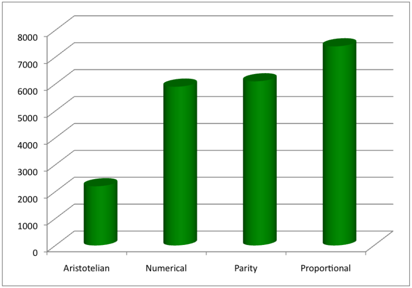
Quantifier group	Examples	Percent
Aristotelian FO	all, some	99
Parity	odd, even	91
Cardinal FO	less than 8, more than 7	92
Proportional	less than half, more than half	85

The percentage of correct answers

Analysis of RT

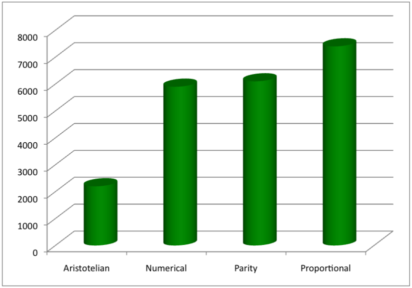


Analysis of RT



RT determined by quantifier type:

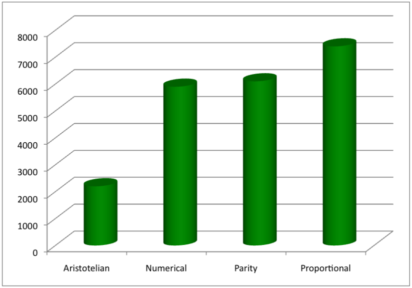
Analysis of RT



RT determined by quantifier type:

- All differences significant;

Analysis of RT



RT determined by quantifier type:

- All differences significant;
  - Aristotelian,
  - parity,
  - cardinal,
  - proportional.

Szymanik & Zajenkowski, Comprehension of Simple Quantifiers. Empirical Evaluation of a Computational Model, Cognitive Science, 2010

Outline

Outline

McMillan et al. fMRI Studies

Differences in brain activity.

McMillan et al. fMRI Studies

Differences in brain activity.

- All quantifiers are associated with numerosity: recruit right inferior parietal cortex.
- Only higher-order activate working-memory capacity: recruit right dorsolateral prefrontal cortex.

McMillan et al. fMRI Studies

Differences in brain activity.

- All quantifiers are associated with numerosity: recruit right inferior parietal cortex.
- Only higher-order activate working-memory capacity: recruit right dorsolateral prefrontal cortex.

But definability seems not to be fine grained enough!

McMillan et al., Neural Basis for Generalized Quantifiers Comprehension, 2005

Szymanik, A Note on Some Neuroimaging Study of Natural Language Quantifiers Comprehension, Neuropsychologia, 2007

Baddeley's Model

WM unified system responsible for the performance in complex tasks.

Baddeley's Model

WM unified system responsible for the performance in complex tasks.

- The model consists of:
  - temporary storage units:
    - phonological loop;
    - visual loop;
  - a controlling system (central executive).

Baddeley, Working Memory and Language: An Overview, 2003

Span Test

Span Test

- To asses the working memory construct.

Span Test

- To asses the working memory construct.
- Subjects read sentences.

Span Test

- To asses the working memory construct.
- Subjects read sentences.
- They are asked to:
  - remember the final words.
  - comprehend the story.

Span Test

- To asses the working memory construct.
- Subjects read sentences.
- They are asked to:
  - remember the final words.
  - comprehend the story.
- What is:
  - the number of correctly memorized words?
  - the degree of understanding?

Span Test

- To asses the working memory construct.
- Subjects read sentences.
- They are asked to:
  - remember the final words.
  - comprehend the story.
- What is:
  - the number of correctly memorized words?
  - the degree of understanding?
- Engagement of processing and storage functions.

Daneman and Carpenter, Individual Differences in Working Memory, 1980

‘Computational’ Theory of WM

Observation

A trade-off between processing and storage functions.

‘Computational’ Theory of WM

Observation

A trade-off between processing and storage functions.

Hypothesis

One cognitive resource – competition for a limited capacity.

Daneman and Merikle, Working Memory and Language Comprehension, 1996

Experimental Setup

Question

How additional memory load influences quantifier verification?

Experimental Setup

Question

How additional memory load influences quantifier verification?

Combined task:

- memorize sequences of digits;
- verify quantifier sentences;
- recall digits.

Predictions

Difficulty (RT and accuracy) should decrease as follows:

- proportional quantifiers,
- numerical quantifiers of high rank,
- parity quantifiers,
- numerical quantifiers of low rank.

Predictions

Difficulty (RT and accuracy) should decrease as follows:

- proportional quantifiers,
- numerical quantifiers of high rank,
- parity quantifiers,
- numerical quantifiers of low rank.

Additionally:

- processing of the PQs should influence storage functions;
- the effect should be stronger in more demanding situation.

Sentence Verification

64 grammatically simple propositions in Polish, like:

1. More than 7 cars are blue.
2. An even number of cars is yellow.
3. Less than half of the cars are black.

Sentence Verification

64 grammatically simple propositions in Polish, like:

1. More than 7 cars are blue.
2. An even number of cars is yellow.
3. Less than half of the cars are black.

- 8 different quantifiers divided into four groups.

Sentence Verification

64 grammatically simple propositions in Polish, like:

1. More than 7 cars are blue.
2. An even number of cars is yellow.
3. Less than half of the cars are black.

- 8 different quantifiers divided into four groups.
    1. numerical quantifiers of relatively low rank, NQ4/5;

Sentence Verification

64 grammatically simple propositions in Polish, like:

1. More than 7 cars are blue.
2. An even number of cars is yellow.
3. Less than half of the cars are black.

- 8 different quantifiers divided into four groups.
  1. numerical quantifiers of relatively low rank, NQ4/5;
  2. numerical quantifiers of relatively high rank, NQ7/8;

Sentence Verification

64 grammatically simple propositions in Polish, like:

1. More than 7 cars are blue.
2. An even number of cars is yellow.
3. Less than half of the cars are black.

- 8 different quantifiers divided into four groups.
  1. numerical quantifiers of relatively low rank, NQ4/5;
  2. numerical quantifiers of relatively high rank, NQ7/8;
  3. parity quantifiers, DQ;

Sentence Verification

64 grammatically simple propositions in Polish, like:

1. More than 7 cars are blue.
2. An even number of cars is yellow.
3. Less than half of the cars are black.

- 8 different quantifiers divided into four groups.
    1. numerical quantifiers of relatively low rank, NQ4/5;
    2. numerical quantifiers of relatively high rank, NQ7/8;
    3. parity quantifiers, DQ;
    4. proportional quantifiers, PQ.

Memory Task

- At the beginning of each trial a sequence of digits.

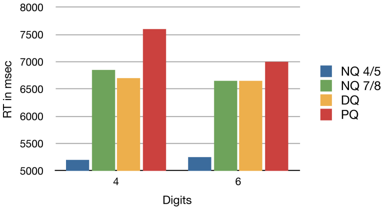
Memory Task

- At the beginning of each trial a sequence of digits.
- 2 experimental conditions:
  - 4 digits
  - 6 digits

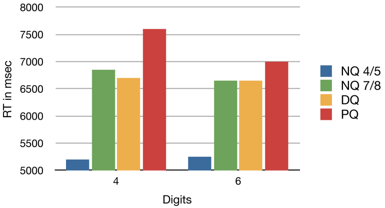
Memory Task

- At the beginning of each trial a sequence of digits.
- 2 experimental conditions:
  - 4 digits
  - 6 digits
- After verification task: recall the string.

RT in Verification Task

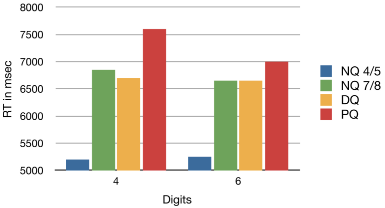


RT in Verification Task



RT determined by quantifier type in 4-digit:

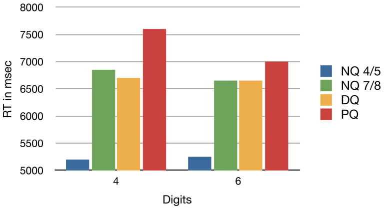
RT in Verification Task



RT determined by quantifier type in 4-digit:

- PQ solved longer than others;

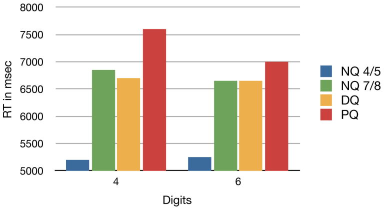
RT in Verification Task



RT determined by quantifier type in 4-digit:

- PQ solved longer than others;
- NQ 4/5 processed shorter than the rest;

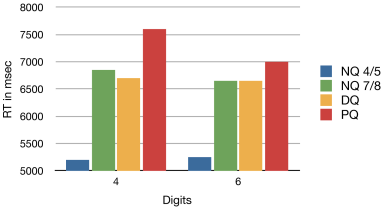
RT in Verification Task



RT determined by quantifier type in 4-digit:

- PQ solved longer than others;
- NQ 4/5 processed shorter than the rest;
- No difference between DQ and NQ 7/8.

RT in Verification Task

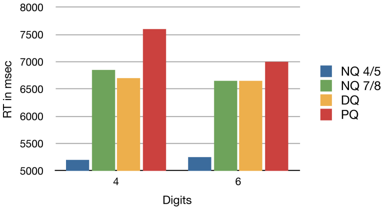


RT determined by quantifier type in 4-digit:

- PQ solved longer than others;
- NQ 4/5 processed shorter than the rest;
- No difference between DQ and NQ 7/8.

6-digit condition:

RT in Verification Task



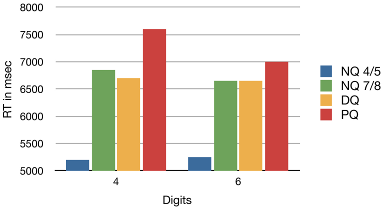
RT determined by quantifier type in 4-digit:

- PQ solved longer than others;
- NQ 4/5 processed shorter than the rest;
- No difference between DQ and NQ 7/8.

6-digit condition:

- NQ 4/5 had the shortest average RT.

RT in Verification Task



RT determined by quantifier type in 4-digit:

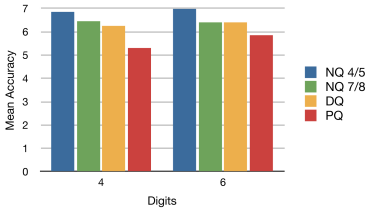
- PQ solved longer than others;
- NQ 4/5 processed shorter than the rest;
- No difference between DQ and NQ 7/8.

6-digit condition:

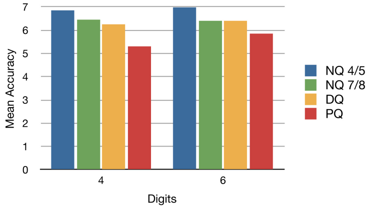
- NQ 4/5 had the shortest average RT.

Only PQ differed between memory load conditions.

Accuracy in Verification Task

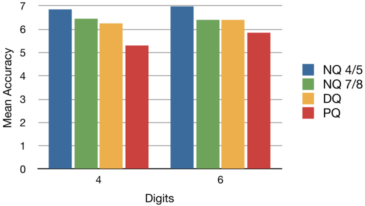


Accuracy in Verification Task



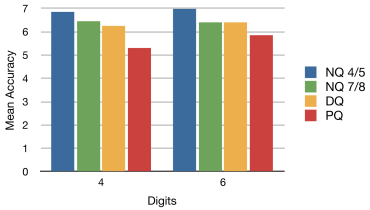
- All quantifiers differed significantly,
- besides DQ and NQ 7/8.

Accuracy in Verification Task



- All quantifiers differed significantly,
- besides DQ and NQ 7/8.
- Large effect for PQ!

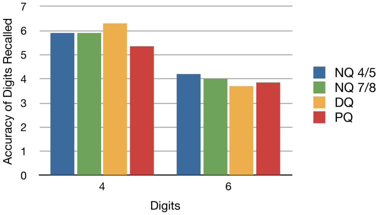
Accuracy in Verification Task



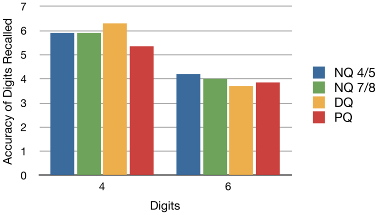
- All quantifiers differed significantly,
- besides DQ and NQ 7/8.
- Large effect for PQ!

Subjects performed worse in 4-digit condition.

Memory Task: Recall Accuracy

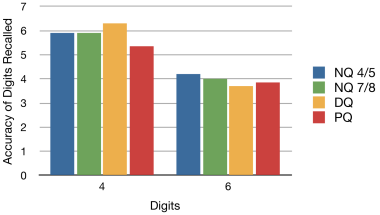


Memory Task: Recall Accuracy



- In 4-digit with PQ: the worst;

Memory Task: Recall Accuracy



- In 4-digit with PQ: the worst;
- In 6-digit: no differences.

Summary

- In 4-digit automata were good predictors of difficulty.

Summary

- In 4-digit automata were good predictors of difficulty.
- Discrepancy under two memory load conditions:
  - The real differences occurred only in 4-digit condition.
  - Holding six elements in memory was probably too difficult.
  - Trade-off between processing and storage.

Summary

- In 4-digit automata were good predictors of difficulty.
- Discrepancy under two memory load conditions:
  - The real differences occurred only in 4-digit condition.
  - Holding six elements in memory was probably too difficult.
  - Trade-off between processing and storage.
- Number of states is a good predictor of the cognitive load.

Szymanik & Zajenkowski,Quantifiers and Working Memory, LNCS, 2010

Proportional Quantifiers

- 4-digit strings accompanying this class were recalled worst.
- But no differences in 6-digit condition:
  - RT decreased: subjects ignored recalling.
- WM engagement PQ processing is qualitatively different.

Outline

Processing vs. Storage

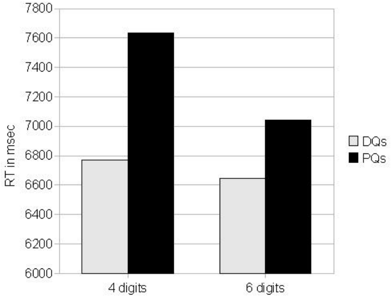
Hypothesis

- *loops := processing;*
- *stack := comparision + storage;*

Question

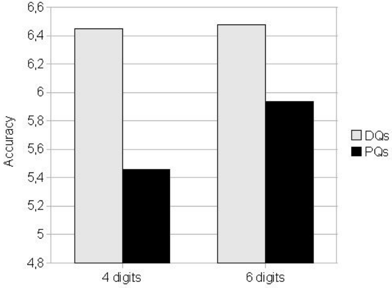
What influences more: processing or storage?

Dual Task: RT

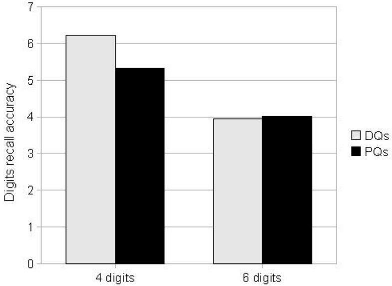


7800
7600
7400
7200
7000
6800
6600
6400
6200
6000
RT in msec
4 digits
6 digits
DQs
PQs

Dual Task: Accuracy



Dual Task: Recall Accuracy



Summary

- Proportional are processed poorer than parity.
- Parity were processed equally under both conditions.
- Proportional in 6: accuracy increased and RT decreased.
- 4 digits were recalled worse with proportional.
- No difference between quantifiers while storing 6 elements.

Summary

- Proportional are processed poorer than parity.
- Parity were processed equally under both conditions.
- Proportional in 6: accuracy increased and RT decreased.
- 4 digits were recalled worse with proportional.
- No difference between quantifiers while storing 6 elements.
- Holding 6 digits in memory universally too difficult.
- So they focused on quantifiers;
- It didn't influence parity;
- But did influenced proportional.

WM in Parity and Proportional Judgments

- Short-term memory retention in verification:
  - Sternberg's STM test;
  - Verification task;

Sternberg's Test

On each trial of the test, the subjects were presented with a random series of different digits, one at a time, for 300 ms, followed by a blank screen and the test digit. Participants had to decide whether the test digit had appeared in the previously displayed string.

Results

- Only in 8 digit condition:
  1. Quantifiers were positively associated with memory task;
  2. The higher memory score, the better verification;
  3. The correlations were very similar.

Memory task	Quantifier accuracy		Quantifier RT	
	DQs	PQs	DQs	PQs
four items	0.18	0.21	0.24	0.24
six items	0.12	0.77	-0.25	-0.05
eight items	0.34*	0.38*	-0.16	0.03

Szymanik & Zajenkowski, Contribution of Working Memory in the Parity and Proportional Judgments, Belgian Journal of Linguistics, forthcoming

Comparison and not Memorization is Really Hard

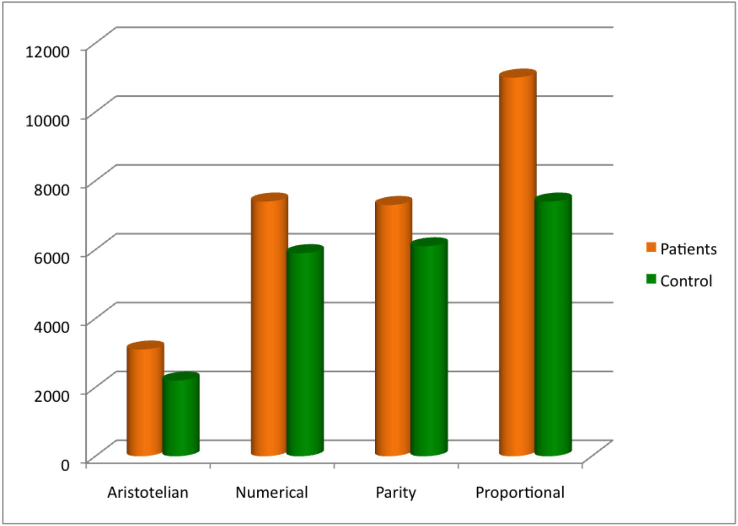
- Proportional and parity quantifiers differ in dual task;
- But they seem to engage STM equally;
- The difference due to the executive resources.

Outline

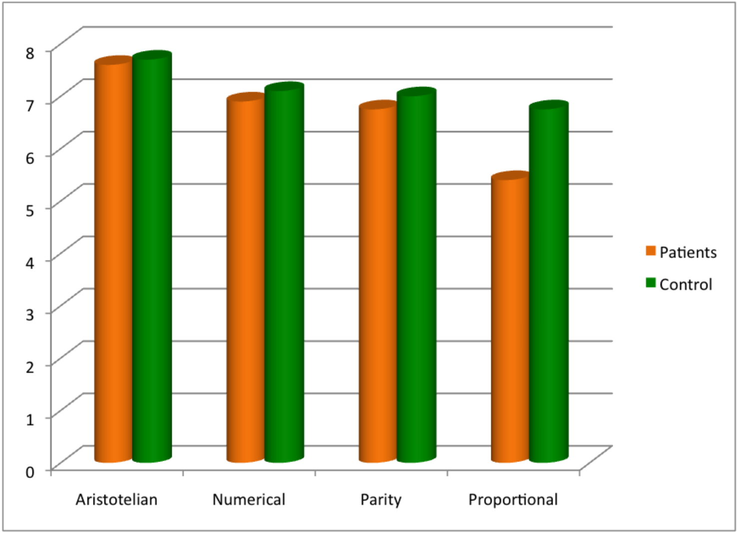
Further Evidence

- Compare performance of:
  - Healthy subjects and
  - Patients with schizophrenia.
- Known WM deficits.

RT Data


12000
10000
8000
6000
4000
2000
0
Aristotelian
Numerical
Parity
Proportional
Patients
Control

RT Data



8
7
6
5
4
3
2
1
0
Aristotelian
Numerical
Parity
Proportional
Patients
Control

Summary

Conclusion

Automata model is psychologically plausible.

Summary

Conclusion

Automata model is psychologically plausible.

Conclusion

Computational complexity $\approx$ cognitive difficulty.

Summary

Conclusion

Automata model is psychologically plausible.

Conclusion

Computational complexity $\approx$ cognitive difficulty.

- As far as we know this is the first empirical proof.
- Between Marr's level 1 and 2.

Outline

Bigger Picture

- Enrich the model:
  1. Approximate Number System;
  2. Visual clues;

Neurocognitive computational modeling

- Mechanism selection;
- Translate to neurocognitive setting;
- fMRI experiments.

Modeling example